Automatic Data Quality Enhancement with Expert Knowledge for Mobile Crowdsensing

Jinghan Jiang*, Kui Wu*, Huan Wang*, Rong Zheng†
*Department of Computer Science, University of Victoria, Canada
†Department of Computing and Software, McMaster University, ON, Canada
email: {jinhanj, wkui, huanwang}@uvic.ca, {rzheng}@mcmaster.ca

***Abstract*—Mobile crowdsensing (MCS) has recently found many applications in environmental monitoring and large-scale surveillance by recruiting crowd workers for data collection and labeling. The quality of labelled data from unknown crowd workers, however, is hard to guarantee. Therefore, it is critical to design a mechanism that can automatically make correct decisions from diverse and even conflicting labels from the crowd. To tackle the challenge, we propose a new algorithm, EFusion, which infuses knowledge from domain experts by asking them to check a small number of labels from the crowd. Taking advantage of cheaper but unreliable crowd workers as well as expensive but reliable experts, EFusion can greatly improve the accuracy in discovering the ground truth of classification-based mobile crowdsensing tasks. EFusion utilizes a probabilistic graphical model and the expectation maximization (EM) algorithm to infer the most likely expertise level for each crowd worker, the difficulty level of tasks, and the ground truth answers. EFusion has been evaluated using real-world case study as well as simulations. Evaluation results demonstrate that EFusion can return more accurate and stable classification results than the majority voting method and state-of-the-art methods.**

I. Introduction

A. Motivation

With technological advances in mobile devices such as smart phone, wearable devices and in-vehicle sensors, mobile crowdsensing (MCS), a special crowd sourcing paradigm that uses the information from a large number of sensing devices and human intelligence to solve difficult problems, has attracted unprecedented interest. MCS has been used in a variety of applications, including environmental monitoring, infrastructure monitoring, and social sensing [1], [2], [3], [4]. Traditionally, MCS recruits a crowd of mobile users to capture data of interest and input their own judgment (i.e., intelligence) to facilitate the processing of big data from the crowd. The quality of data/judgment from a user obviously depends on whether or not the user is knowledgeable about the subject matter. Accordingly, there have been research efforts in matching qualified users and given tasks [5].

Recently, there is a surge of interest in another form of MCS where the sensing crowd consists of "smart" devices/programs that possess machine intelligence. One example is the smart cameras that recognize human faces or detect urgent events such as a car collision. Another example is WeChat mini program *pet recognition*, which can tell, with a level of confidence, the breeds of dogs or cats from the pictures taken with phones. Those smart devices/programs have a certain level of intelligence, and as such they can be treated as another source of information critical to MCS applications.

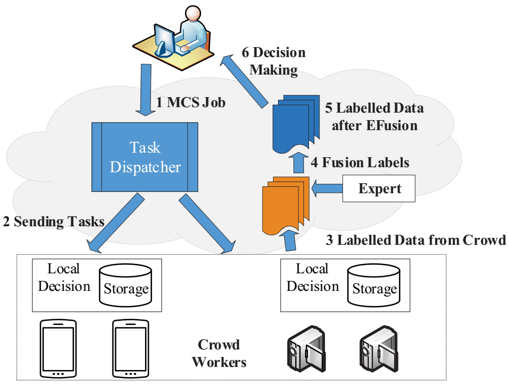


Fig. 1: EFusion in the work flow of MCS.

Take MCS to detect distracted driving as an example. A typical work flow is as follows: An end user (e.g., police department) submits an MCS job (e.g., detecting distracted driving at a busy intersection during a time period). The MCS platform uses a task dispatcher [6], [7] to assign sensing tasks to mobile users or smart surveillance cameras, which are called **crowd workers** or simply **workers** of MCS. After receiving the tasks, the crowd workers capture pictures of distracted drivers (i.e., labelled data with crowd workers' local judgments), and the end user makes final decisions based on the labelled data from crowd workers. The typical work flow of MCS consists of Steps 1, 2, 3, and 6, as shown in Fig. 1.

In the typical work flow of MCS as above, a core challenge is that the labelled data from the crowd may be error prone. The correctness of answers from humans is subject to their domain knowledge; the accuracy of answers from smart devices/programs is limited by their lower computational power and storage. In addition, optimized models [8] might not be easy to implement with smart devices; hence the accuracy of their answers is generally inferior than that obtained in cloud data centers. Therefore, MCS needs a "quality assurance" mechanism to verify the results from crowd workers. Clearly, depending on the professionals to monitor and check *all* results from crowd workers is prohibitive and defeats the original purpose of using cheap crowd workers. The question we will answer in this paper is: *how can we infuse the knowledge* *learned with a small amount of reliable labels from the professionals to automatically correct crowd workers' labels?*

B. Overview of the Proposed Solution

Let **Expert** be a group of professionals who have expert domain knowledge. Here we do not distinguish individual experts in this paper, and use Expert to denote them as a whole. Answers from Expert are assumed to be more accurate than those from crowd workers, and the knowledge from Expert could help us make better decisions. Following this idea, we propose a new algorithm, *EFusion*, which infuses the knowledge from Expert into the unreliable data from the crowd to automatically identify and eliminate incorrectly labelled data. The work flow of MCS with EFusion (Steps 4, 5) is shown in Fig. 1.

Since it is unrealistic to ask Expert to answer all questions due to the higher cost (e.g., higher hourly rate) for professionals, EFusion takes advantage of both Expert and crowd workers. For the labelled data from crowd workers with *unknown* expertise levels, we assign a small portion of data to Expert and ask Expert to judge the labels. The answers from Expert are explored to infer the most likely expertise level for each crowd worker, as well as the ground-truth answers.

The contributions of the paper includes:

- We design a new algorithm, EFusion, which utilizes the knowledge from Expert to gauge answers from unreliable distributed devices and workers. By taking advantage of both types of contributors, EFusion greatly improves the likelihood in discovering the ground truth from MCS.
- We solve the non-trivial inference problem in EFusion, which infers the ground truth labels, the expertise level of each contributing smart device/crowd worker, and the difficulty level of questions.
- We perform a thorough evaluation of EFusion using real-world case study as well as simulations. Evaluation results demonstrate that EFusion outperforms other popular methods, such as majority voting, the DS method [9], the method for Conflict Resolution on Heterogeneous Data (CRH) [10], and the Generative model of Labels, Abilities, Difficulties (GLAD) [11].

II. Related Work

Existing work can be roughly divided into two categories: discovering truth in MCS and utilizing expert knowledge to improve quality of answers from crowd sourcing.

In the first category, the goal of proposed solutions is to handle the situation where the ground truth is unknown and data contributed from multiple workers in MCS are inconsistent or even contradictory. Wang et al. [5] used the EM algorithm to determine whether one user's answer can be accepted as the truth. Peng et al. [12] extended the work in [5] by establishing a connection between the quality of user's sensing data and their reward. Liu et al. [13] estimated the truth based on the estimation of data quality from different users in an online manner. To get the accurate estimation of user's reliability, a model which combine multiple properties is also proposed in [10]. In a more specific setting where there are correlations among monitored entities, Meng et al. [14] formulated an optimization problem to find truth. Under the same assumption, Wang et al. [15] provided a scalable approach that exploits dependencies between observed variables to improve fact-finding accuracy of social sensing data.

In the second category, the problem of infusing professional knowledge into crowd workers has been investigated [16] [8]. The key idea is to combine answers from different groups (workers and experts). This concept has been used in active learning to obtain the effective model on truth discovery [17] [8]. Tang et al. [18] proposed a semi-supervised approach to combining the labels from experts and workers so that the consensus labels can be inferred. Sheshadri and Lease [19] provided an open source shared task framework to compare the performance of various statistical consensus methods. Both solutions assumed that experts always know the ground truth. In addition to these works, Aroyo and Welty [20] aggregated the labels from experts and workers by k-score to train a model for semantic interpretation of sentences.

EFusion belongs to the second category. However, the purpose and usage of experts labels in EFusion are different from existing methods. In active learning, the purpose of incorporating expert opinions is to select the most informative questions for participants to label, so that a better classification model could be trained. In our algorithm, however, the expert's opinion is used to directly infer more truthful labels. Furthermore, it can *automatically* infer the ground truth and expertise level of workers in MCS. Meanwhile, the Expert opinion in our algorithm does not necessarily represent the ground truth, while in others methods it does.

III. Details of EFusion

A. Problem Formulation

We consider a batch of m classification tasks in MCS to be labeled. To improve the quality of answers from workers, we try to gain some expertise knowledge on the batch by asking Expert to label a (small) subset of k instances, where $k < m$. Since the focus is on crowd workers' reliability, we do not distinguish individual experts in this paper, and use **Expert** to denote them as a whole. Let the total number of crowd workers be n.

Denote the answer of crowd worker i for question j by $l_{ij}, 1 \leq i \leq n, 1 \leq j \leq m$. Denote Expert's answers as $e_l, 1 \leq l \leq k$. For ease of presentation, we assume that l_{ij} and e_l are binary values, while the model can be easily extended to multi-class classification tasks, as discussed later in Section III-E. $l_{ij} = 0$ means user i gives the wrong label to question j, otherwise $l_{ij} = 1$. Similarly, $e_l = 0$ if the Expert gives wrong label to question l, else $e_l = 1$ Note that $l_{ij}(1 \leq i \leq n, 1 \leq j \leq m)$ and $e_l(1 \leq l \leq k)$ are inputs to EFusion.

Given the batch of questions, we assume that Expert has a higher chance of giving correct answers (answers are closer to the ground truth) than crowd workers. We thus introduce the concept of *expertise level*: the higher the expertise level, the higher the probability of returning a correct answer. Associated with a crowd worker i is the expertise level

$\alpha_i \in (-\infty, +\infty)$, and associated with Expert is the expertise level $\alpha_E \in (0, +\infty)$, where $+\infty$ means that the labeler always answers correctly and $-\infty$ means that the labeler always answers incorrectly. A positive/negative expertise value implies the labeler is more likely to return a correct/incorrect answer for a specific question. We assume *a priori distribution* for α_i and assume a priori value for α_E, which is larger than the prior mean of α_i. Since the difficulty level of a question may impact the probability that a correct answer can be obtained, we introduce a positive parameter $\beta_j (1 \leq j \leq m)$ to denote the difficulty level of question j. We assume *a priori distribution* for β_j. $\beta_j \in (0, +\infty)$, where a small β_j means that it is easier to answer the question correctly by the same worker. Note that the difficulty of a question, *by definition*, is considered to be a feature of the question and independent of the crowd workers. The likelihood of a truthful answer from worker i to question j is jointly decided by i's expertise level and j's difficulty level.

The notations used in the paper are listed in Table I.

Problem 1. *Given the probabilistic graphical model in Figure 2 and the observed values $l_{ij} (1 \leq i \leq n, 1 \leq j \leq m)$ and $e_l (1 \leq l \leq k)$, what are the ground-truth answers $Z_j (1 \leq j \leq m)$? what are the posteriori estimates of α_i? what are the posteriori estimates of β_j?*

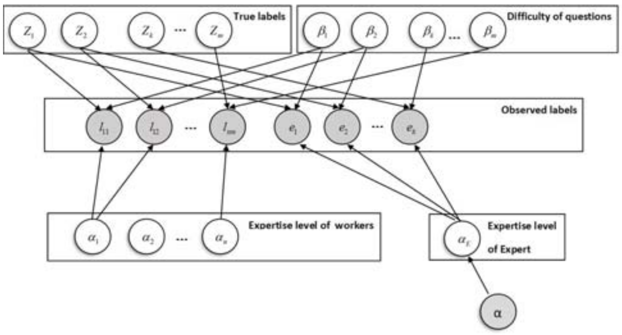


Fig. 2: Graphical model of Z_j, α_i, α_E, l_{ij}, β_j and e_j. Note that only the shaded part are observed.

B. Probability of Correct Answers

In general, the more difficult a question, the lower the probability that a correct answer can be obtained; and the higher the expertise level, the higher the probability that a correct answer can be obtained. As such, we propose to model the probability that a correct answer is returned using a logistic function. It is worth noting that the logistic function and its variants have been widely used in financial domain for calculating probability of correct prediction [21] as well as in similar problem settings [11].

In particular, the probability that Expert returns the correct answer to question j is modeled as:

$$p(e_j = Z_j | Z_j, \alpha_E, \beta_j) = \frac{1}{1 + e^{-\alpha_E / \beta_j}}, j \in \{1, \ldots, k\}. \quad (1)$$

The probability that crowd worker i returns the correct answer to question j is modeled as:

$$p(l_{ij} = Z_j | Z_j, \alpha_i, \beta_j) = \frac{1}{1 + e^{-\alpha_i / \beta_j}}. \quad (2)$$

TABLE I: Summary of main notations

Notation	*Description*
α_i	the expertise level of worker i
l_{ij}	the answer worker i gives for question j
α_E	the expertise level of Expert
β_j	the difficulty of question j
n	the number of workers
m	the number of questions in the batch
k	the number of questions answered by Expert
Z_j	the ground truth answer for question j
e_l	the answer of Expert for question l

C. Main Algorithm

We use the Expectation Maximization (EM) [22] algorithm to solve Problem 1. To simplify notation, we denote $\bar{\boldsymbol{\alpha}}_W = \{\alpha_1, \ldots, \alpha_n\}$, $\bar{\boldsymbol{\alpha}} = \bar{\boldsymbol{\alpha}}_W \cup \{\alpha_E\}$. Starting with initial prior values of α_1, …, α_n, α_E, β_1, …, β_m, the EM algorithm iterates through the following E step and M step.

1) The E Step: Given observed values of $\{l_{11}, \ldots, l_{nm}\}$ and $\{e_1, \ldots, e_k\}$, we first calculate the posterior probabilities of all true answers Z_j using the estimated $\bar{\boldsymbol{\alpha}}$ and β_j's from the last M step. Denote $l_{.j} = \{l_{1j}, \ldots, l_{nj}\}$. Without loss of generality, we assume the first k questions are answered by Expert as well as crowd workers. For each $j \leq k$, we have

$$\begin{aligned} & p(Z_j | l_{.j}, e_j, \bar{\boldsymbol{\alpha}}, \beta_j) \\ & \propto p(Z_j) p(e_j | Z_j, \alpha_E, \beta_j) \prod_i^n p(l_{ij} | Z_j, \bar{\boldsymbol{\alpha}}_W, \beta_j) \end{aligned} \quad (3)$$

where $p(Z_j | \bar{\boldsymbol{\alpha}}, \beta_j) = p(Z_j)$ because of the conditional independence assumptions from the graphical model. Similarly, for each question $j (k < j \leq m)$ answered only by crowd workers, we have:

$$\begin{aligned} & p(Z_j | l_{.j}, \bar{\boldsymbol{\alpha}}, \beta_j) \\ & \propto p(Z_j) \prod_i^n p(l_{ij} | Z_j, \bar{\boldsymbol{\alpha}}_W, \beta_j) \end{aligned} \quad (4)$$

2) The M Step: The auxiliary function Q is defined as the expectation of joint log likelihood of the observed and hidden variables ($l_{.j}$,Z_j), given the parameters ($\bar{\boldsymbol{\alpha}}_W$,$\beta_j$).

$$\begin{aligned} & Q(\bar{\boldsymbol{\alpha}}_W, \beta_j) \\ =& E[\ln \prod_{j=1}^{k} p(e_j, l_{.j}, Z_j | \bar{\boldsymbol{\alpha}}, \beta_j) \prod_{j=k+1}^{m} p(l_{.j}, Z_j | \bar{\boldsymbol{\alpha}}, \beta_j)] \\ =& \sum_{j=k+1}^{m} E[\ln p(Z_j)] + \sum_{i=1,j=1}^{i=n,j=k} E[\ln p(l_{ij} | Z_j, \bar{\boldsymbol{\alpha}}_W, \beta_j)] + \\ & \sum_{j=1}^{k} E[\ln p(Z_j) p(e_j | Z_j, \alpha_E, \beta_j)] + \\ & \sum_{i=1,j=k+1}^{i=n,j=m} E[\ln p(l_{ij} | Z_j, \bar{\boldsymbol{\alpha}}_W, \beta_j], \end{aligned} \quad (5)$$

where the expectation is computed from the posterior probabilities in the E step. Let α^p, β^p denote the α and β estimated by the previous iteration. The first part of Equation (5) can be expanded to Equation (6) as follows:

$$\begin{aligned}&\sum_{j=k+1}^{m} E[\ln p(Z_j)]\\&=\sum_{j=k+1}^{m} (p(Z_j=0|\mathbf{L},\alpha^p,\beta^p)\ln p(Z_j=0)+\\&p(Z_j=1|\mathbf{L},\alpha^p,\beta^p)\ln p(Z_j=1)),\end{aligned} \tag{6}$$

in which $\mathbf{L}$ means all the labels given for the $(k+1)$-th to m-th questions.

The second part and the last part of Equation (5) have similar form (i.e., the only difference is on the range of j value) and thus can be expanded in similar form shown in Equation (7):

$$\begin{aligned}&\sum_{ij} E[\ln p(l_{ij}|Z_j,\bar{\boldsymbol{\alpha}}_W,\beta_j]\\&=\sum_{ij}(p(Z_j=0|\mathbf{L},\alpha^p,\beta^p)\ln p(l_{ij}|Z_j=0,\alpha_i,\beta_j)+\\&p(Z_j=1|\mathbf{L},\alpha^p,\beta^p)\ln p(l_{ij}|Z_j=1,\alpha_i,\beta_j)),\end{aligned} \tag{7}$$

where $p(l_{ij}|Z_j=0,\alpha_i,\beta_j)$ and $p(l_{ij}|Z_j=1,\alpha_i,\beta_j)$ can be attained with Equation (1). Without causing confusion, we here slightly abuse the notation by using $\mathbf{L}$ to denote all the labels given for the questions in the same range as that of j.

The third part of Equation (5) can be expanded to Equation (8):

$$\begin{aligned}&\sum_{j=1}^{k} E[\ln p(Z_j)p(e_j|Z_j,\alpha_E,\beta_j)]\\&=\sum_{j=1}^{k}(p(Z_j=0|\mathbf{L},\alpha^p,\beta^p)\ln p(e_j|Z_j=0,\alpha_E,\beta_j)+\\&p(Z_j=1|\mathbf{L},\alpha^p,\beta^p)\ln p(e_j|Z_j=1,\alpha_E,\beta_j)),\end{aligned} \tag{8}$$

where Equation (1) is used to calculate $p(e_j|Z_j=0,\alpha_E,\beta_j)$ and $p(e_j|Z_j=1,\alpha_E,\beta_j)$.

Then we use gradient descent to find the values of $\bar{\boldsymbol{\alpha}}_W,\beta_j$ to maximize the function Q. Note that we assume the value of α_E is known and thus we do not update α_E in the EM algorithm. The algorithm iterates through the E step and the M step until convergence. Here, convergence means that either the number of iterations reaches a given maximum threshold or the difference in learned parameters between consecutive iterations falls within a given small threshold.

The posterior probabilities of Z_j values are obtained after the **last** (i.e., the one before the algorithm stops) E step. After the last E step, for each question $j (1 \le j \le m)$, we use Equation (1) and Equation (2) to calculate the probability of getting correct labels from workers and Expert, respectively, and then choose the label with the highest overall probability as the final label for this question.

D. Priors on Parameters

Both $\bar{\boldsymbol{\alpha}}_W$ and β_j are continuous random variables, and we used Gaussian priors on $\bar{\boldsymbol{\alpha}}_W$, and truncated Gaussian priors on β_j's such that all β_j values are positive. In particular, we assume that the expertise level of each worker follows a normal distribution $\mathcal{N}(\mu_1,\sigma_1^2)$, and the difficulty of each question follows a truncated normal distribution $\mathcal{N}(\mu_2,\sigma_2^2)$. Note that Gaussian prior is a mathematically convenient and practically reasonable assumption. The expertise level of Expert is assumed to be known in advance, and is denoted as $\alpha_E=\alpha$. We set $\alpha >> \mu_1$ since Expert has much more expertise than crowd workers on average. The value of prior probabilities of each class (i.e., $p(Z_j)$) is also influential on the performance of EFusion. So the task publishers who have some domain knowledge on the assigned tasks can acquire better estimated parameters by changing $p(Z_j)$ in the E-step.

E. Further Discussion

EFusion can be easily extended to handle scenarios involving multi-class classification. Suppose that answers are grouped into S categories. Given a question j, let its correct answer be Z_j. The probabilities that Expert and worker i gives the correct answer can still be calculated with Equation (1) and Equation (2), respectively. It is reasonable to assume incorrect answers are equally probable. In other words, we have:

$$\begin{aligned}&p(e_j=s'|Z_j,\alpha_E,\beta_j)=\frac{e^{-\alpha_E/\beta_j}}{(S-1)(1+e^{-\alpha_E/\beta_j})},\\&\quad j\in\{1,\ldots,k\},s'\neq Z_j.\end{aligned} \tag{9}$$

$$\begin{aligned}&p(l_{ij}=s'|Z_j,\alpha_i,\beta_j)=\frac{e^{-\alpha_i/\beta_j}}{(S-1)(1+e^{-\alpha_i/\beta_j})},\\&\quad i\in\{1,\ldots,n\},j\in\{1,\ldots,m\},...\end{aligned} \tag{10}$$

Consequently, in the M step of EFusion, Equation (9) and Equation (10) should be used to calculate $p(e_j|Z_j,\alpha_E,\beta_j)$ and $p(l_{ij}|Z_j,\bar{\boldsymbol{\alpha}}_W,\beta_j)$ in Equation (5), and the Q value in each iteration. The rest of EFusion remains unchanged.

IV. Evaluation

A. Baseline Methods

For comparison, we implemented the following truth discovery methods:

- Majority voting (MV): The final answer to a question is the answer that appears the most of times among all crowd workers. And all contributors are treated equally, so there is no "Expert".
- The DS method [9]: It uses full confusion matrices to denote the expertise of each contributor, the EM algorithm is used to obtain maximum likelihood estimates of ground truth of polytomous classes problem under medical background.
- Conflict Resolution on Heterogeneous Data (CRH) [10]: This is a general model for truth discovery from multiple sources that might have different data types. It uses an optimization framework where truths and source reliability are defined as two sets of unknown variables, with the objective to minimize the overall weighted deviation between the truths and the multi-source observations where each source is weighted by its reliability.

- Generative model of Labels, Abilities, Difficulties (GLAD) [11]: GLAD makes decisions regarding ground truth, difficult level of questions, and expertise level of workers, using a similar graphical model as in EFusion but without any inputs from Expert.

In the datasets used in the case studies and simulations, the ground-truth answer of each question is given. This allows us to compute exactly the estimate errors of EFusion and other baseline methods. We adopt the following measures to evaluate the performance of different methods.

- Accuracy: It is defined as the ratio of correct answers from different methods over the total number of questions in the batches.
- F-measure: It is the harmonic mean of precision and recall, where precision is the proportion of predicted positive labels and real positives labels, and recall is the proportion of real positive labels that are correctly predicted positive.

B. Case Study

We carry out a MCS case study, detection of distracted driving. As discussed in the motivating example, distracted driving is a main cause of accidents in our daily transportation. Even if many cities have law enforcement, most distracted drivers remain uncaught due to the high cost in detecting distracted driving. One solution is to launch a MCS campaign by recruiting volunteers or using smart cameras on streets to report distracted drivers and upload images during a certain time period.

To *emulate* a MCS campaign, we use a public dataset [23] which consists of 606 photos, each recording a potential distracted driver. The images and the ground truth labels can be found from [23]. To emulate the action of workers, we posted the labelling task with a fee in CrowdFlower [24] and asked crowd workers to determine whether or not the driver in each of the 606 photos is fatigue or distracted. In this way, we collected 15150 answers in total from 25 workers, and apply different methods to determine the final answers.

We set $p(Z_j)$ to 0.5, β_j to 1 for $j = 1, \dots, m$, where $m = 606$. We set $\alpha_E = 3$ so that Expert has around 90% chance to give a correct answer for each question. Since we know the ground-truth labels, answers from Expert are simulated by returning the correct answers with 0.9 probability. The results for the case study are summarized in Fig. 3, where Fig. 3(a) and Fig. 3(b) demonstrate the performance of different methods with different numbers of workers in terms of accuracy and F-measure. In the figures, EFusion is shortened as EF. Note that we only draw the performance of EFusion when Expert answers 40% questions, and use an error bar to present the accuracy and F-measure achieved by EFusion when the percentage of questions answered by Expert changes from 20% to 80%. The precision and recall of different methods are summarized in Table II.

As shown in Figure 3(a) and Figure 3(b), EFusion outperforms all the baseline methods for different numbers of workers. On average, EFusion has about 10% improvement over the best baseline CRH in terms of accuracy and F-measure. Majority voting performs the worst among all the methods in this case study. Notably, both the accuracy and F-measure of CRH remain the same as the number of workers changes. This is because the performance of CRH is mainly subject to the data heterogeneity, while in this case only one type of categorical data is involved. According to CRH frame, the algorithm can infer a part of reliable workers, and then detect truth only based on them. Note that having more workers does not help for all schemes in this case. This is because when the accuracy is around fifty-fifty, it appears roughly half of the workers give the correct results regardless the total number of workers.

We are interested in studying the trade off between accuracy and knowledge infusion. As such, we introduce *coverage rate of Expert*, defined as the percentage of questions answered by Expert, to capture knowledge infusion. The results are presented in Fig. 3(c). We can see that as expected, the more questions answered by Expert, the more accurate the final results. With more questions answered by Expert (e.g., higher than 70%), the improvement in final accuracy diminishes. In other words, the marginal utility of infusing expert knowledge decreases. Such a relationship can be used to guide decisions on how much expertise knowledge is needed. Furthermore, EFusion has indeed utilized the knowledge from crowd workers as the accuracy is consistently better than that relying on Expert alone (e.g., 0.9 x coverage rage).

Since in the empirical studies we do not know the ground truth about workers expertise and the difficulty levels of questions, we cannot evaluate the performance of EFusion with respect to the accuracy of inferred workers expertise and questions difficulty levels. Next, we perform controlled simulations to evaluate these two aspects.

C. Simulation Evaluation

In the simulation, 2000 questions are generated in total, each having a binary answer. For ground-truth label of each question, we set its value to 0 or 1 randomly with equal probability. A total of 25 workers were simulated.

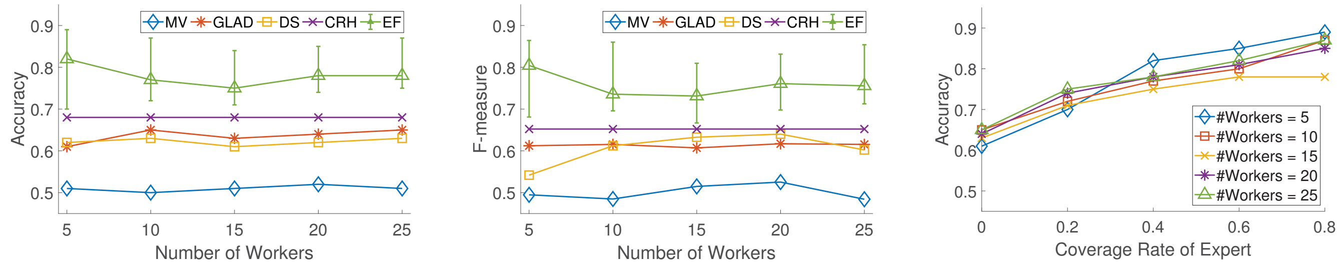
We first simulate a base case where majority voting can achieve a reasonably good accuracy. We set the ground-truth values of workers expertise α_W following normal distribution $\mathcal{N}(\mu = 1, \sigma = 0.2)$ and set the value of Expert expertise α_E to $5(\gg \bar{\alpha}_W)$. The difficulty levels of the 2000 questions in the batch are drawn from an independent truncated normal distribution $\mathcal{N}(\mu = 5, \sigma = 1)$. For each parameter setting, the simulation is repeated 50 times to smooth out the variability among trails. In each trail, we computed the accuracy, F-measure for all methods, as well as the correlation between the estimated expertise level, the estimated question difficulty and their ground truth. The results in all the figures of this section reflect the mean values from the 50 trails.

The accuracy and F-measure with 5 to 25 workers and the accuracy under different Expert's coverage rates are summarized in Fig. 4. The simulation results further confirm that: 1) For the accuracy and F-measure of the five methods, EFusion outperforms all the baselines under different numbers

TABLE II: Performance Comparison on Precision and Recall Under Different Number of Workers

		#Workers=5		**#Workers=10**		**#Workers=15**		**#Workers=20**		**#Workers=25**	
		Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Method	*MV*	0.4528	0.5455	0.4364	0.5455	0.4561	0.5909	0.4727	0.5909	0.4510	0.5227
	GLAD	0.5099	0.5359	0.4864	0.5957	0.6462	0.5725	0.6591	0.5800	0.6364	0.5957
	DS	0.5524	0.5318	0.6818	0.5556	0.7045	0.5741	0.7273	0.5714	0.6364	0.5714
	CRH	0.6818	0.6250	0.6818	0.6250	0.6818	0.6250	0.6818	0.6250	0.6818	0.6250
	EF(20%)	0.7273	0.6400	0.7273	0.6667	0.7027	0.5909	0.7143	0.6818	0.7209	0.7045
	EF(40%)	0.8409	0.7708	0.7442	0.7273	0.7727	0.6939	0.7955	0.7292	0.7727	0.7391
	EF(60%)	0.8864	**0.7959**	0.8409	0.7400	0.7805	0.7273	**0.8636**	0.7451	**0.8864**	0.7500
	EF(80%)	**0.9459**	0.7955	**0.9091**	**0.8163**	**0.8500**	**0.7727**	0.8409	**0.8222**	0.8444	**0.8636**

Note: the value in the parentheses after "EF" denotes the coverage rate of Expert in EFusion.



(a) Accuracy of different methods on detection of distracted driving.

(b) F-measure of different methods on detection of distracted driving.

(c) Performance of EFusion on detection of distracted driving when varying the Expert coverage rate.

Fig. 3: The comparison of different methods when $\alpha_W \sim \mathcal{N}(\mu = 1, \sigma = 0.2)$, $\beta_j \sim \mathcal{N}(\mu = 5, \sigma = 1)$, $\alpha_E = 5$.

of workers. 2) As the coverage rate of Expert increases, EFusion achieves higher accuracy with different number of workers. Detailed simulation results are given in Table III.

To evaluate the accuracy, we compute the root-mean-square error (RMSE) between the worker's expertise (/question's difficulty levels) estimated by EFusion and the corresponding true values. Fig. 4(d) shows the RMSE with different numbers of workers. It can be seen that both the estimated workers' expertise and the estimated question difficulty are closer to the corresponding true values as the number of workers increases.

To investigate the stability of EFusion, we simulate more difficult settings where distributions of α_W and β_j vary. On the basis of original distribution, we increase the variance of α_W, increase the variance of β_j, decrease the mean value of α_W to negative, and increase the mean value of β_j in turn. Fig. 5 shows the performance of baseline methods and EFusion under different α_W, β_j settings, using 40% coverage rate of Expert. Fig. 5(a) and Fig. 5(b) show the results when the numbers of workers are 5 and 25, respectively. From the figure, it can be observed that EFusion gets higher accuracy than the baseline methods under different settings when the number of workers varies from 5 to 25. The performance of EFusion remains stable even if α_W and/or β_j have a high variance.

V. Conclusions

Crowdsensing that uses the information from crowd workers in finding answers from various sensing data has been applied into more and more areas. Observing that domain experts can return more reliable answers compared to the crowd, we design a model, EFusion, that infuses the domain knowledge into crowd's answers so that the false answers could be corrected *automatically*. Meanwhile, the expertise levels of crowd workers and the difficulty level of questions will also be inferred. The results demonstrate that in addition to the high robustness and good performance in estimating parameters, EFusion outperforms all the baseline methods in terms of accuracy and f-measure.

References

[1] P. Dutta, P. M. Aoki, N. Kumar, A. Mainwaring, C. Myers, W. Willett, and A. Woodruff, "Common sense: participatory urban sensing using a network of handheld air quality monitors," in *Proceedings of the 7th ACM conference on embedded networked sensor systems*, 2009, pp. 349–350.

[2] X. Mao, X. Miao, Y. He, X.-Y. Li, and Y. Liu, "Citysee: Urban co 2 monitoring with sensors," in *IEEE INFOCOM*, 2012, pp. 1611–1619.

[3] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE Communications Magazine*, vol. 49, no. 11, 2011.

[4] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T. Campbell, "Bikenet: A mobile sensing system for cyclist experience mapping," *ACM Transactions on Sensor Networks (TOSN)*, vol. 6, no. 1, p. 6, 2009.

[5] D. Wang, L. Kaplan, H. Le, and T. Abdelzaher, "On truth discovery in social sensing: A maximum likelihood estimation approach," in *Proceedings of the 11th international conference on Information Processing in Sensor Networks*, 2012, pp. 233–244.

[6] W. Gong, B. Zhang, and C. Li, "Task assignment in mobile crowdsensing: present and future directions," *IEEE Network*, no. 99, pp. 1–8, 2018.

[7] S. Yangy, K. Hany, Z. Zhengy, S. Tangz, and F. Wu, "Towards personalized task matching in mobile crowdsensing via fine-grained user profiling," in *IEEE Conference on Computer Communications*, 2018, pp. 1–9.

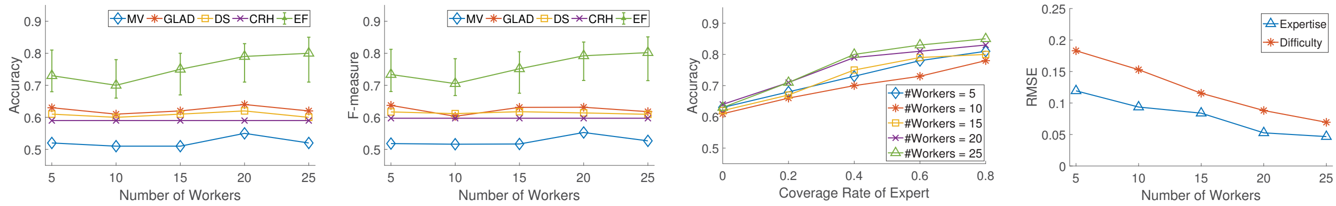
[8] A. T. Nguyen, B. C. Wallace, and M. Lease, "Combining crowd and expert labels using decision theoretic active learning," in *Third AAAI Conference on Human Computation and Crowdsourcing*, 2015.

[9] A. P. Dawid and A. M. Skene, "Maximum likelihood estimation of observer error-rates using the em algorithm," *Applied statistics*, pp. 20–28, 1979.

TABLE III: Performance Comparison on Precision and Recall Under Different Number of Workers in Simulation

		#Workers=5		#Workers=10		#Workers=15		#Workers=20		#Workers=25	
		Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Method	*MV*	0.5049	0.532	0.501	0.5322	0.5127	0.5209	0.544	0.5616	0.5225	0.5308
	GLAD	0.6385	0.636	0.6117	0.5949	0.636	0.6262	0.643	0.6024	0.6008	0.6356
	DS	0.6193	0.6145	0.6227	0.6008	0.6189	0.6164	0.6385	0.591	0.6182	0.6106
	CRH	0.5996	0.5949	0.5996	0.5949	0.5996	0.5949	0.5996	0.5949	0.5996	0.5949
	EF(20%)	0.6945	0.6673	0.6693	0.6667	0.6792	0.6712	0.7179	0.7123	0.7188	0.7104
	EF(40%)	0.7996	0.728	0.7081	0.7025	0.7636	0.7397	0.8016	0.7828	0.8116	0.7926
	EF(60%)	0.7867	0.7836	0.7415	0.7241	0.7652	0.7505	0.8216	0.8023	0.8415	0.8285
	EF(80%)	**0.8024**	**0.8043**	**0.7893**	**0.7769**	**0.8063**	**0.8031**	**0.8415**	**0.8285**	**0.8632**	**0.8395**

Note: the value in the parentheses after "EF" denotes the coverage rate of Expert in EFusion.



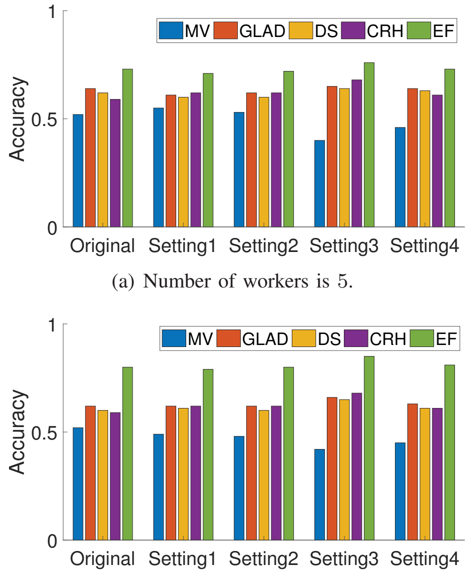
(a) Accuracy of different methods in the simulation.

(b) F-measure of different methods in the simulation.

(c) Performance comparison of EFusion in the simulation when varying the Expert coverage rate.

(d) RMSE between estimated parameters and the corresponding true values.

Fig. 4: The performance of different methods and the RMSE between estimated parameters and the corresponding true values in the simulation when $\alpha_W \sim \mathcal{N}(\mu = 1, \sigma = 0.2)$, $\beta_j \sim \mathcal{N}(\mu = 5, \sigma = 1)$, $\alpha_E = 5$.



(a) Number of workers is 5.

(b) Number of workers is 25.

Fig. 5: Performance of different methods in various settings: Original: $\alpha_W \sim \mathcal{N}(\mu = 1, \sigma = 0.2)$, $\beta_j \sim \mathcal{N}(\mu = 5, \sigma = 1)$
1: $\alpha_W \sim \mathcal{N}(\mu = 1, \sigma = 2)$, $\beta_j \sim \mathcal{N}(\mu = 5, \sigma = 1)$
2: $\alpha_W \sim \mathcal{N}(\mu = 1, \sigma = 0.2)$, $\beta_j \sim \mathcal{N}(\mu = 5, \sigma = 5)$
3: $\alpha_W \sim \mathcal{N}(\mu = -1, \sigma = 0.2)$, $\beta_j \sim \mathcal{N}(\mu = 5, \sigma = 1)$
4: $\alpha_W \sim \mathcal{N}(\mu = 1, \sigma = 0.2)$, $\beta_j \sim \mathcal{N}(\mu = 10, \sigma = 1)$.

[10] Q. Li, Y. Li, J. Gao, B. Zhao, W. Fan, and J. Han, "Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation," in *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, 2014, pp. 1187–1198.

[11] J. Whitehill, T.-f. Wu, J. Bergsma, J. R. Movellan, and P. L. Ruvolo, "Whose vote should count more: Optimal integration of labels from labelers of unknown expertise," in *Advances in neural information processing systems*, 2009, pp. 2035–2043.

[12] D. Peng, F. Wu, and G. Chen, "Pay as how well you do: A quality based incentive mechanism for crowdsensing," in *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2015, pp. 177–186.

[13] S. Liu, Z. Zheng, F. Wu, S. Tang, and G. Chen, "Context-aware data quality estimation in mobile crowdsensing," in *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*. IEEE, 2017, pp. 1–9.

[14] C. Meng, W. Jiang, Y. Li, J. Gao, L. Su, H. Ding, and Y. Cheng, "Truth discovery on crowd sensing of correlated entities," in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, 2015, pp. 169–182.

[15] S. Wang, L. Su, S. Li, S. Hu, T. Amin, H. Wang, S. Yao, L. Kaplan, and T. Abdelzaher, "Scalable social sensing of interdependent phenomena," in *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*, 2015, pp. 202–213.

[16] N. Q. V. Hung, N. T. Tam, L. N. Tran, and K. Aberer, "An evaluation of aggregation techniques in crowdsourcing," in *International Conference on Web Information Systems Engineering*. Springer, 2013, pp. 1–15.

[17] Y. Bachrach, T. Graepel, T. Minka, and J. Guiver, "How to grade a test without knowing the answers—a bayesian graphical model for adaptive crowdsourcing and aptitude testing," *arXiv preprint arXiv:1206.6386*, 2012.

[18] W. Tang and M. Lease, "Semi-supervised consensus labeling for crowdsourcing," in *SIGIR 2011 workshop on crowdsourcing for information retrieval (CIR)*, 2011, pp. 1–6.

[19] A. Sheshadri and M. Lease, "Square: A benchmark for research on computing crowd consensus," in *First AAAI Conference on Human Computation and Crowdsourcing*, 2013.

[20] L. Aroyo and C. Welty, "The three sides of crowdtruth," *Journal of Human Computation*, vol. 1, pp. 31–34, 2014.

[21] G. Creamer and S. Stolfo, "A link mining algorithm for earnings forecast and trading," *Data mining and knowledge discovery*, vol. 18, no. 3, pp. 419–445, 2009.

[22] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[23] K. Diaz-Chito, A. Hernández-Sabaté, and A. M. López, "A reduced feature set for driver head pose estimation," *Applied Soft Computing*, vol. 45, pp. 98–107, 2016.

[24] *Crowdflower*, https://www.crowdflower.com/.